

Modellierung und Darstellung von Prozessvarianten in Provop

Alena Hallerbach¹, Thomas Bauer¹ und Manfred Reichert²

¹Daimler AG Forschung und Technologie, Deutschland

{alena.hallerbach; thomas.tb.bauer}@daimler.com

²Institut f. Datenbanken und Informationssysteme, Universität Ulm, Deutschland
manfred.reichert@uni-ulm.de

Zusammenfassung: Die Praxis hat gezeigt, dass man es bei der Modellierung von Prozessen oftmals mit zahlreichen Varianten zu tun hat. Jede Prozessvariante stellt dabei eine Anpassung an bestimmte Rahmenbedingungen dar. Heutige Modellierungswerkzeuge ermöglichen lediglich das Ausmodellieren von Prozessvarianten in separaten Prozessmodellen, woraus ein hoher Wartungsaufwand resultiert: Werden z.B. grundlegende Prozessanpassungen erforderlich (z.B. infolge gesetzlicher Änderungen), sind meist mehrere Varianten betroffen und deshalb bei diesem Ansatz auch mehrere Prozessmodelle anzupassen. Dies wiederum führt schnell zu Inkonsistenzen oder Fehlern. Der vorliegende Beitrag greift diese Problemstellungen auf. Er stellt den Provop-Lösungsansatz zur Modellierung mehrerer Prozessvarianten in einem Prozessmodell, mit expliziter Kennzeichnung der verschiedenen Varianten, vor. Damit lassen sich die Aufwände für die Modellierung und Wartung von Prozessvarianten signifikant reduzieren sowie Inkonsistenzen bei der Anpassung von Varianten vermeiden.

1 Motivation

Arbeitsabläufe in großen Unternehmen und Organisationen werden oftmals mittels Prozessmodellen dokumentiert, für deren Erstellung verschiedene Modellierungswerkzeuge wie ARIS Business Architect [IDS06] oder WBI Modeler [IBM07] zum Einsatz kommen. Ein einzelnes Prozessmodell beschreibt üblicherweise einen bestimmten Ablauftyp (z.B. Kreditantragstellung, Reisekostenabrechnung) und legt fest, welche Aktivitäten in welcher Reihenfolge und mit welchen Ressourcen durchzuführen sind. Durch Dokumentation der Arbeitsabläufe in Prozessmodellen werden diese für die Prozessbeteiligten transparent und können z.B. durch Analyse und Simulation verbessert werden [Sch98, Gad05].

Fallstudien haben gezeigt, dass Prozesse in der Praxis häufig in unterschiedlichen Varianten auftreten (siehe auch [VDA05, Mey96]). Im Automobil-Bereich etwa finden sich mehrere Dutzend Prozessvarianten in der Produktentwicklung, deren konkrete Festlegung abhängig vom zu entwickelnden Produkt (z.B. PKW oder LKW) sowie von organisatorischen Zuständigkeiten, strategischen Ausrichtungen und sonstigen Rahmenbedingungen erfolgt. Abbildung 1 zeigt beispielhaft den vereinfachten Prozess zur Abwicklung

von Produktänderungsanträgen: Nachdem ein Änderungsantrag angestoßen wurde, gibt der Prozessverantwortliche Stellungnahmen verschiedener Bereiche in Auftrag. Sind diese Stellungnahmen nicht konfliktfrei, werden sie vom Projektleiter zu einer Gesamtstellungnahme zusammengefasst. Der Änderungskreis, d.h. das Entscheidungsgremium für Änderungen, entscheidet dann über die Genehmigung oder Ablehnung des Änderungsantrags. Abschließend erfolgt die Umsetzung der beantragten Produktänderung.

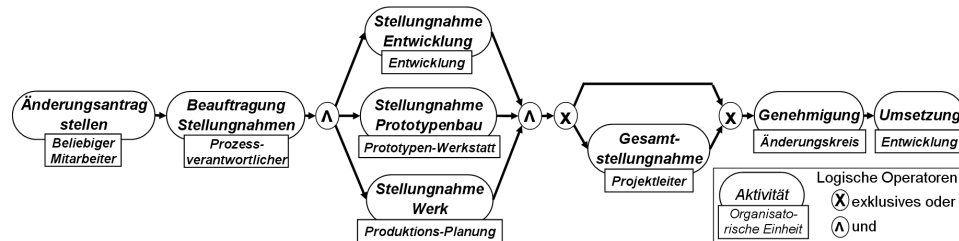


Abbildung 1: Vereinfachter Prozess zur Abwicklung eines Änderungsantrags

In der Praxis wird dieser Prozess unter verschiedenen Rahmenbedingungen ausgeführt, die jeweils eine Abweichung vom hier dargestellten Ablauf erforderlich machen. Dies wiederum führt zur Entstehung von Prozessvarianten: Steigen z.B. die Qualitätsanforderungen an das Produkt, wird eine zusätzliche Stellungnahme der Abteilung für Qualitätssicherung während der Abwicklung des Produktänderungsantrags notwendig. Es ist auch möglich, dass organisatorische Verantwortlichkeiten neu zugewiesen werden müssen. Soll z.B. die Bearbeitungszeit der Stellungnahmen auf Grund hohen Zeitdrucks verkürzt werden, so kann der Bereich „Entwicklung“ die Verantwortung zur Erstellung der Stellungnahmen selbst übernehmen und die ursprünglich verantwortlichen Fachbereiche anschließend über die geschätzten Kosten informieren.

Obwohl die Modellierung solcher Prozessvarianten eine hohe Praxisrelevanz hat, bieten existierende Prozessmodellierungswerkzeuge keine adäquate Unterstützung. Sie erfordern im Prinzip das Ausmodellieren jeder einzelnen Prozessvariante in einem separaten Prozessmodell. Liegen für einen Prozess viele Varianten vor¹, gelangt der Modellierer schnell an den Punkt, an dem die Vielzahl der resultierenden Prozessmodelle für ihn nicht mehr handhabbar ist. Des Weiteren steigen Kosten und Aufwände für die Wartung, da bei grundlegenden Änderungen (z.B. infolge gesetzlicher Anpassungen) jedes Prozessmodell separat angepasst werden muss. Damit einher geht eine hohe Fehleranfälligkeit, die zu inkonsistenten Prozessmodellen führt.

Aufgrund der skizzierten Unzulänglichkeiten und Anforderungen an das Management von Prozessvarianten [HBR08], entwickeln wir mit Provop (PROzessVarianten mittels OPTio-nen) einen Ansatz, der es erlaubt, die Variantenvielfalt von Prozessen in einem gemeinsamen Prozessmodell abzubilden: Für alle Prozessvarianten wird ein Prozess festgelegt, der als Ausgangspunkt für die Definition der Varianten dient. Die Varianten werden dann in Form von variantenspezifischen Abweichungen modelliert und explizit im Prozessmodell des Ausgangsprozesses vorgehalten. Auf diese Weise sollen Kosten und Aufwand

¹ Je nach Anwendungsfall können dies einige Dutzend bis zu mehreren hundert Prozessvarianten sein.

für die Erstellung und Wartung von Prozessmodellen deutlich reduziert werden. Eine übersichtliche Darstellung aller Varianten bzw. aller variantenspezifischen Abweichungen erhöht zudem die Vergleichbarkeit und Analysierbarkeit.

Kapitel 2 beschreibt die grundlegende Idee von Provop und dient der Begriffsklärung. In Kapitel 3 werden Operationen zur Modellierung variantenspezifischer Abweichungen vorgestellt. Kapitel 4 erläutert grundlegende Aspekte der Darstellung von Prozessvarianten aus Modellierersicht. Nach der Vorstellung von verwandten Arbeiten in Kapitel 5 schließt der Beitrag mit einer Zusammenfassung und einem Ausblick auf unsere zukünftigen Arbeiten in Kapitel 6.

2 Definition von Prozessvarianten mittels Optionen

Soll für einen Prozess eine neue Variante definiert werden, erzeugt man diese mit heutigen Modellierungswerkzeugen meist durch Kopieren und Anpassen eines Ausgangsprozesses. Die resultierende Prozessvariante weist dann typischerweise eine hohe Ähnlichkeit mit diesem Ausgangsprozess auf. Diese Ähnlichkeit kann für die Modellierung von Prozessvarianten genutzt werden: Ausgehend von einem *Basisprozess*, für den Varianten gebildet werden sollen, werden die variantenspezifischen Prozessanpassungen definiert und zusammen mit dem Basisprozess in einem Modell integriert.

Eine Möglichkeit, diesen Ansatz zu verfolgen, besteht in der Modellierung der verschiedenen Prozessvarianten mittels bedingter bzw. alternativer Verzweigungen. Vereinfacht ausgedrückt repräsentiert dann jeder mögliche Pfad durch das Prozessmodell eine Variante. Durch Festlegung geeigneter Verzweigungsbedingungen, die Bezug auf Prozessdaten nehmen können, wird dann die jeweils gewünschte Prozessvariante determiniert. Ein Nachteil dieses nahe liegenden Modellierungsansatzes ist, dass Prozessvarianten nicht explizit als solche gekennzeichnet sind. Vielmehr ist die Information darüber, wann eine bestimmte Variante durchlaufen wird, in der Ablauflogik des Prozessmodells versteckt. Insbesondere kann der Modellierer nicht mehr erkennen, ob eine bedingte Verzweigung im Prozessmodell eine Variante oder aber eine „normale“ Verzweigung darstellt, die bei allen Varianten auftritt und deshalb Teil des Basisprozesses ist. Aber nicht nur dem Anwender geht bei diesem Ansatz das Modellierungswissen über Varianten verloren, sondern auch dem System selbst. Dies hat zur Folge, dass bei der Darstellung des Prozessmodells keine Unterscheidung der Prozessvarianten und auch keine Hervorhebung bzw. kein Ausblenden einzelner Prozessvarianten möglich ist.

Um alle Informationen von Prozessvarianten explizit in einem Modell zu definieren, verwenden wir in Provop eigenständige Objekte, die sogenannten *Optionen*. Diese beschreiben die variantenspezifischen Anpassungen eines Basisprozesses. Die Anwendung einer oder mehrerer Optionen auf den Basisprozess führt dann zu einem *Ergebnismodell*, d.h. einem Prozessmodell, das eine konkrete Prozessvariante beschreibt. Abbildung 2 zeigt die einzelnen Provop-Komponenten. Sie werden im Folgenden im Detail vorgestellt.

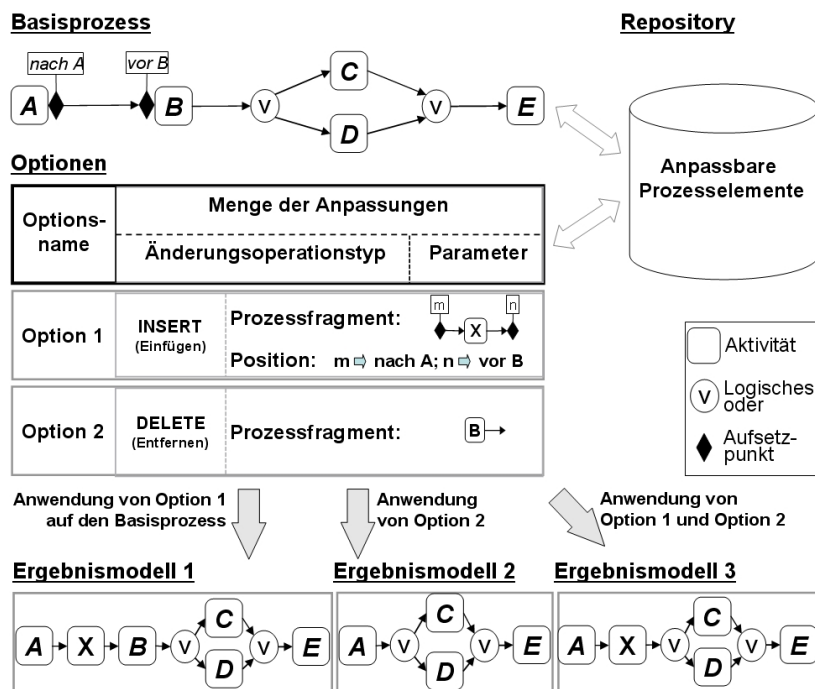


Abbildung 2: Die Komponenten des Provop-Ansatzes

2.1 Basisprozess

Wie erwähnt werden in Provop die Prozessvarianten ausgehend von einem *Basisprozess* modelliert. Dieser Basisprozess kann prinzipiell nach unterschiedlichen Aspekten festgelegt werden:

- **Standardprozess:** Verwendung eines domänenspezifischen Standardprozesses als Basisprozess.
- **„Häufigster“ Prozess:** Ein Prozess, der im Vergleich zu anderen Prozessvarianten besonders häufig verwendet wird, kann ebenfalls als Basisprozess dienen. Abweichungen im Sinne von Prozessvarianten werden seltener benötigt.
- **Minimaler Modellierungsaufwand:** Der Basisprozess wird so gewählt, dass der Modellierer zur Varianten-Bildung möglichst wenige Anpassungen vornehmen muss.
- **Schnittmenge aller Prozessvarianten:** Der Basisprozess wird als Schnittmenge aller Prozessvarianten modelliert. Das heißt, jedes Element des Basisprozesses ist in allen abgeleiteten Varianten vorhanden. Bei der Anpassung des Basisprozesses zwecks Variantenbildung ist dann kein Element aus dem Basisprozess zu entfernen.

Die vorgestellten Aspekte zur Festlegung eines Basisprozesses unterscheiden sich in einem weiteren Punkt: Standardprozesse und häufig verwendete Prozesse beschreiben spezifische Anwendungsfälle. Dagegen sind Basisprozesse, die als Schnittmenge aller Varianten definiert werden, oder die für einen minimalen Modellierungsaufwand optimiert sind, konstruierte Prozesse, die nur der Bildung von Prozessvarianten dienen.

Die Frage, ab wann die Modellierung mit einem zweiten Basisprozess für eine Klasse von Prozessvarianten sinnvoll ist, kann nicht allgemein gültig beantwortet werden. Dies hängt vom spezifischen Anwendungsfall ab. Faktoren sind zum Beispiel, wieviele Prozessvarianten durch Optionen zu modellieren sind und wie stark sich die zu generierenden Prozessvarianten voneinander unterscheiden.

2.2 Optionen

Unser Ziel ist es, die erforderlichen Anpassungen des Basisprozesses bei der Ableitung von Prozessvarianten explizit in das Modell des Basisprozesses zu integrieren. Dazu werden eigenständige Repräsentations-Objekte mit eigener Semantik benötigt, die wir im Folgenden als *Optionen* bezeichnen. Ihre Anwendung im Prozessmodell ist optional, d.h. sie erfolgt abhängig von der aktuell benötigten Prozessvariante. Jede Option folgt dem Schema aus Abbildung 2: Neben dem definierten Basisprozess gibt es eine Menge von Optionen, deren ggf. kombinierte Anwendung auf den Basisprozess zu unterschiedlichen Ergebnismodellen führt. Eine einzelne Option besteht aus einem Optionsnamen, welcher die Option eindeutig identifiziert, und einer Menge von Anpassungen, die durch die Option am Basisprozess vorgenommen werden. Eine einzelne Anpassung wiederum besteht aus einer bestimmten Prozess-Änderungsoperation und der zu ihrer Durchführung notwendigen Parameter. Die möglichen Provop-Änderungsoperationen sind INSERT, DELETE, MOVE und MODIFY. Sie werden in Kapitel 3 erläutert.

Die Elemente eines Basisprozesses, wie Aktivitäten, Kanten und Strukturknoten² werden in einem *Repository* verwaltet. Unter Verwendung ihrer ID können einzelne Prozesselemente von Änderungsoperationen referenziert und angepasst werden. Außerdem kann das Repository mittels INSERT mit weiteren Elementen befüllt werden.

Ein weiteres Mittel zur Identifikation zu ändernder Prozesselemente sind in Provop die sogenannten *Aufsetzpunkte*. Diese repräsentieren sowohl graphisch als auch logisch diejenigen Stellen des Basisprozesses, die von einer bestimmten Änderung betroffen sind. In Provop definieren wir solche Aufsetzpunkte an Aktivitäten und Strukturknoten. Wir unterscheiden dabei zwischen den Ein- und Ausgängen der Elemente, um so die Position der Änderung genau angeben zu können. Im Beispiel aus Abbildung 2 werden für die dargestellte INSERT-Operation die Aufsetzpunkte „nach A“ am Knotenausgang von A und „vor B“ am Knoteneingang von B gewählt. Durch die Verwendung von verschiedenen IDs und Aufsetzpunkten können die Anpassungen einer Option auf den ganzen Basisprozess verteilt wirken.

²Als Strukturknoten werden z.B. Schleifenknoten und Verzweigungs- bzw. Synchronisationsknoten mit logischem Operator bezeichnet.

Innerhalb einer Option werden Aufsetzpunkte verwendet, um z.B. die Ein- und Ausgänge eines einzufügenden Prozessfragments (d.h. einer Menge von Prozesselementen) zu markieren und dann auf die konkreten Aufsetzpunkte im Basisprozess abzubilden (vgl. Option 1 in Abbildung 2). Dieser Schritt macht aus der generisch beschriebenen Option eine spezifische Option für den modellierten Basisprozess. Optionen können auf diese Weise in mehreren Prozessmodellen verwendet werden, indem die Abbildungsbeziehungen der Aufsetzpunkte an den aktuellen Basisprozess angepasst werden.

Es liegen noch keine Erfahrungswerte vor, wieviele Optionen zur Abbildung aller Varianten eines Prozesstyps benötigt werden. Da je nach Anwendungsfall eine bestimmte Anzahl an Prozessvarianten modelliert werden müssen und diese auf unterschiedliche Weise mittels Optionen abgebildet werden können (z.B. eine Option pro Prozessvariante oder eine Option pro Änderungsoperation) variiert die Anzahl der Optionen stark.

2.3 Ableitung von Prozessvarianten

Sind für einen Basisprozess, wie in Abbildung 2 dargestellt, eine oder mehrere Optionen modelliert, können die spezifischen Prozessvarianten durch deren Anwendung abgeleitet werden. Dazu werden die einzelnen Anpassungen einer Option, d.h. die angegebenen Änderungsoperationen mit ihren Parametern, nacheinander auf den Basisprozess angewendet und so das Ergebnismodell erstellt. In Abbildung 2 sind alle Ergebnismodelle, die sich durch Anwendung von Option 1 und/ oder Option 2 ergeben können, dargestellt.

Die zum Erzeugen einer bestimmten Prozessvariante erforderlichen Anpassungen müssen nicht notwendigerweise in einer Option modelliert werden; auch die Kombination mehrerer Optionen zur Ableitung einer Variante ist möglich (vgl. Ergebnismodell 3 in Abbildung 2). Für den Modellierer heißt das, dass er die Anpassungen auf einzelne Optionen verteilen kann. Dies ermöglicht die Wiederverwendung der Optionen, z.B. zur Modellierung „ähnlicher“ Prozessvarianten, die durch Kombination derselben Option mit anderen Optionen entstehen.

Die Kombination von Optionen birgt einige Herausforderungen: So muss definiert werden, wie widersprüchliche Anpassungen (z.B. Löschen und Verändern des gleichen Prozesselements im Basisprozess), redundante Anpassungen (z.B. doppeltes Einfügen des gleichen Elements) und nicht kommutative Anpassungen (d.h. je nach Anwendungsreihenfolge der Optionen ergeben sich unterschiedliche Ergebnismodelle) zu handhaben sind. In Provop befassen wir uns intensiv mit derartigen Fragestellungen. Auf Grund des beschränkten Platzes verzichten wir an dieser Stelle jedoch auf die Vorstellung der detaillierten Problemstellungen und entsprechenden Lösungsansätze.

3 Definition der Anpassungen

In Provop werden dem Modellierer die Änderungsoperationen `INSERT`, `DELETE`, `MOVE` und `MODIFY` zur Definition von Anpassungen innerhalb von Optionen angeboten. Jede

Änderungsoperation stellt, zusammen mit ihrer jeweiligen Parametrisierung, eine konkrete Anpassung eines Basisprozesses dar. Im Folgenden werden die einzelnen Änderungsoperationen vorgestellt. Es werden ihr Anwendungszweck, benötigte Parameter und relevante Problemstellungen erläutert. Aus Platzgründen verzichten wir auf formale Darstellungen.

3.1 Einfügen von Prozessfragmenten

Ein `INSERT` ermöglicht es dem Modellierer, den Basisprozess in verschiedenerlei Hinsicht zu erweitern. So kann er sowohl einzelne Prozesselemente (z.B. Aktivitäten, Kanten und Strukturknoten) als auch ganze Prozessfragmente einfügen.

Notwendige Parameter sind, neben der Angabe was eingefügt werden soll, die Information darüber, *an welcher Stelle* das betreffende Prozessfragment im Basisprozess einzufügen ist. Ein `INSERT` benötigt für diesen Zweck zumindest zwei Aufsetzpunkte, von denen der eine den Startpunkt markiert (Start-Aufsetzpunkt) und der andere das Ende des einzufügenden Prozessfragments (Ende-Aufsetzpunkt). Diese Start- und Ende-Aufsetzpunkte werden innerhalb einer Anpassung zunächst generisch beschrieben (vgl. Abbildung 3), um so die Wiederverwendbarkeit der modellierten Option für andere Basisprozesse zu ermöglichen. Die konkreten Ziel-Aufsetzpunkte an Aktivitäten und Strukturknoten eines bestimmten Basisprozesses werden dann innerhalb der jeweiligen Anpassung paarweise auf die Start- und Ende-Aufsetzpunkte abgebildet.

Abbildung 3 zeigt eine Option des in Abbildung 1 vorgestellten Prozesses für eine Prozessvariante mit erhöhten Qualitätsansprüchen. Diese wird durch ein `INSERT` beschrieben, um eine zusätzlich notwendige Stellungnahme in den Basisprozess einzufügen. Ein Parameter dieser Operation ist das einzufügende Prozessfragment mit der Aktivität *Stellungnahme Qualität*. Des Weiteren legt Option 1 die Aufsetzpunkte „Start“ und „Ende“ fest, die beide durch eine Kontrollfluss-Kante mit der einzufügenden Aktivität verbunden sind. Schließlich ist definiert, dass der Start-Aufsetzpunkt der Option auf den Aufsetzpunkt „Stellungnahmen beauftragt“ im Basisprozess abgebildet wird. Analog ist für den Ende-Aufsetzpunkt der Aufsetzpunkt „Stellungnahmen eingeholt“ angegeben.

Auch das Ergebnismodell, das aus der Anwendung von Option 1 auf den Beispielprozess resultiert, zeigt Abbildung 3: Der zusätzliche Schritt wird an den Aufsetzpunkten als paralleler Pfad in den Basisprozess eingefügt.

Im Allgemeinen führt nicht jedes `INSERT` wieder zu einem korrekten und konsistenten Ergebnismodell. In Provop wird daher sowohl bei der Modellierung der Anpassungen als auch im Anschluss an systemseitige Konsistenzprüfungen auf entsprechende Fehler hingewiesen.

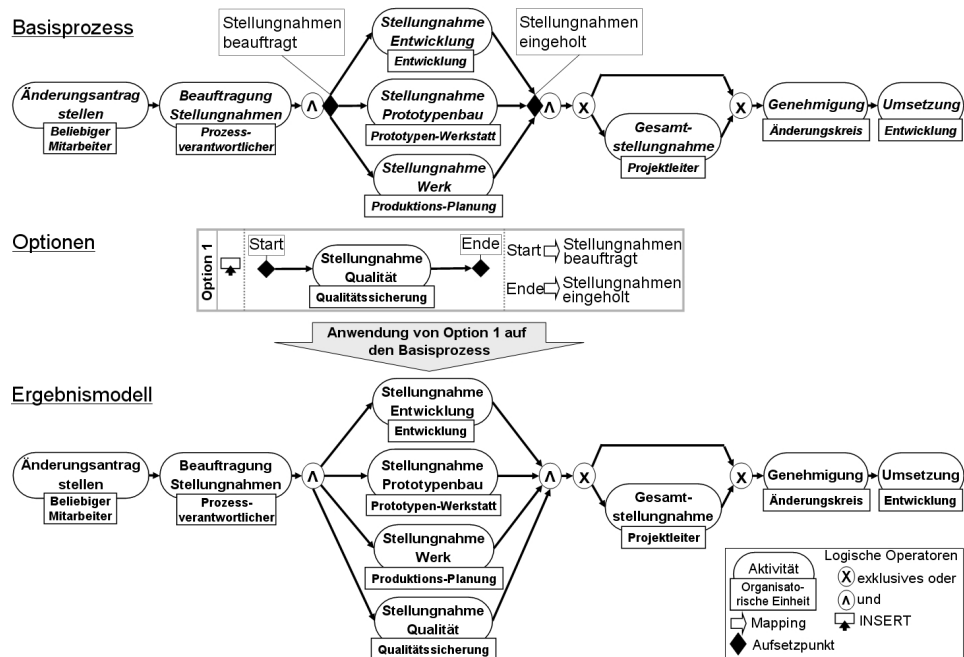


Abbildung 3: Einfügen eines Prozessfragments in den Basisprozess aus Abbildung 1

3.2 Entfernen von Prozessfragmenten

Durch Entfernen von Prozessfragmenten kann der Modellierer den Basisprozess reduzieren. Für ein DELETE müssen die zu entfernenden Prozess-Elemente angegeben werden. Dafür gibt es im Prinzip zwei mögliche Ansätze, die beide von Provop unterstützt werden:

- Angabe von Aufsetzpunkten:** Durch Angabe von Start- und Ende-Aufsetzpunkten kann ein Prozessfragment des Basisprozesses zum Entfernen markiert werden. Nachteilig ist dabei, dass die einzelnen Elemente des Prozessfragments nicht explizit benannt werden. Dies kann ggf. zum ungewollten Entfernen weiterer Elemente führen, wenn andere Optionen ebenfalls Aktivitäten in den angegebenen Bereich einfügt haben.
- Angabe von IDs:** Ein weiterer Ansatz ist das gezielte Entfernen von Elementen durch die Angabe ihrer ID. Dies ist vor allem bei wenig zu entfernenden Elementen sinnvoll. Muss ein größeres Prozessfragment entfernt werden, kann der Modellierer z.B. durch Ziehen eines Auswahl-Rechtecks im Modellierungswerkzeug den Aufwand gering halten. Abbildung 4 zeigt eine Option, die eine Aktivität durch Angabe ihrer ID aus dem Basisprozess entfernt.

Beim Entfernen von Prozessfragmenten ist zu garantieren, dass die resultierende Prozessvariante wieder durch ein korrektes Prozessmodell beschrieben wird. Dazu muss sicherge-

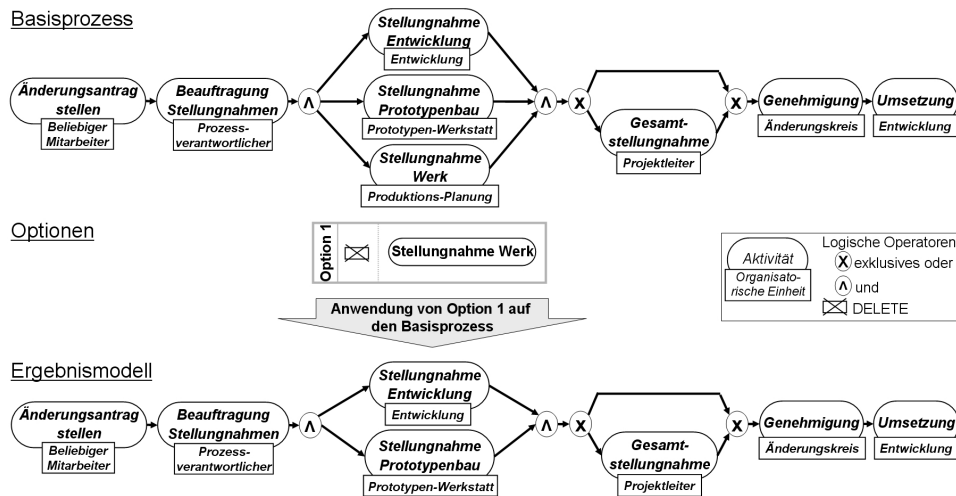


Abbildung 4: Definition einer Prozessvariante durch Entfernen einer Aktivität des Basisprozesses

stellt sein, dass nach dem Entfernen keine offenen Enden bzw. „Prozesslücken“ entstehen. Dies kann z.B. passieren, wenn ein Knoten gelöscht wird, die ein- und ausgehenden Kanten aber ohne neu definierten Ziel- bzw. Startknoten im Prozessmodell verbleiben. Solche Situationen erfordern Konsistenzprüfungen oder müssen, wie in Provop, direkt von der Änderungsoperation korrigiert werden (z.B. indem Prozesslücken automatisch geschlossen werden).

Schließt ein zu löschendes Prozessfragment Aufsetzpunkte im Basisprozess ein, so könnte das Löschen dieser Aufsetzpunkte dazu führen, dass andere Änderungsoperationen nicht mehr korrekt angewendet werden können. Dies kann vermieden werden, indem die Aufsetzpunkte für andere Änderungsoperationen weiterhin vorgehalten werden, z.B. durch den Erhalt der Prozessstruktur mit leeren Knoten oder durch Verschieben der Aufsetzpunkte.

3.3 Verschieben von Prozessfragmenten

Das Verschieben von Prozessfragmenten (MOVE) ermöglicht es dem Modellierer, für eine Prozessvariante die Ausführungsreihenfolge von Aktivitäten des Basisprozesses zu verändern. Dazu gibt er bei der Optionsdefinition die Start- und Ende-Aufsetzpunkte des zu verschiebenden Prozessfragments an; weiter bestimmt er die neue Zielposition im Basisprozess durch Angabe weiterer Aufsetzpunkte. Es muss beachtet werden, dass für Start- und Zielposition jeweils die gleiche Anzahl Aufsetzpunkte spezifiziert wird, damit ein korrektes paarweises Abbilden zwischen den ursprünglichen Ein- und Ausgängen auf die neuen Kantenenden stattfinden kann. Die MOVE-Operation stellt in Provop wieder sicher, dass keine Lücken im Prozessmodell entstehen.

3.4 Ändern von Attributen

Zusätzlich zu den beschriebenen Änderungsoperationen, die Kontrollflussanpassungen im Basisprozess ermöglichen, können mit Hilfe der `MODIFY`-Operation auch Attribute von Prozesselementen angepasst werden. Attribute einer Aktivität sind z.B. Dauer und Bearbeitungsrolle. Darüber hinaus kann durch Ändern der Zuordnung von Aktivitäten zu Datenobjekten die Weitergabe von Prozessdaten (Datenfluss) angepasst werden. Bei Strukturknoten können Schleifen- bzw. Verzweigungsbedingungen an einer logischen OR-/ XOR-Verzweigung verändert werden. Bestimmte Attribute sollen nicht geändert werden können, wie z.B. die ID eines Prozesselements, um ungewünschte Seiteneffekte durch falsche Referenzierungen zu vermeiden. Provop verwendet dazu *fixe*, d.h. unveränderliche Attribute. Sie dürfen von einer `MODIFY`-Operation nicht angepasst werden.

All diese Änderungen werden mit Hilfe der ID eines Prozesselements und der Attributbezeichnung realisiert. Hinsichtlich Konsistenz ist zu beachten, dass die ID auch tatsächlich an ein Prozesselement vergeben ist und der referenzierte Elementtyp (Aktivität, Kante oder Strukturknoten) dem erwarteten entspricht.

4 Darstellung von Varianten im Modellierungswerkzeug

4.1 Angezeigte Informationen von Optionen

Ein variantenbehaftetes Prozessmodell sollte auch bei hoher Anzahl von Optionen übersichtlich und verständlich bleiben. In Provop wird der Modellierer deshalb durch angemessene Darstellungsformen unterstützt, von denen er für die jeweilige Aufgabenstellung die für ihn passende wählen kann. Im Folgenden werden exemplarisch drei von Provop unterstützte Darstellungskonzepte für Optionen vorgestellt:

- **Vollständige Anzeige der Optionen:** Abbildung 5a zeigt ein Beispiel, bei dem alle Informationen von Option 1 und Option 2 vollständig angezeigt werden. Diese Darstellung erlaubt dem Modellierer eine detaillierte Sicht auf die von den einzelnen Optionen vorgenommenen Anpassungen.
- **Minimalisierte Anzeige der Optionen:** Im Beispiel aus Abbildung 5a werden für Option 3 nur wenige Informationen angezeigt. Dies können, wie im Beispiel dargestellt, die Bezeichnung der Option und die zugehörigen Änderungsoperationen sein. Auch eine benutzerdefinierte Auswahl der angezeigten Daten ist möglich.
- **Teilweise minimalisierte Anzeige der Optionen:** Die Kombination aus den Konzepten „Vollständige Anzeige“ und „Minimalisierte Anzeige“ erlaubt dem Modellierer das Ein- und Ausblenden von Details. Das bedeutet, dass ein Wechsel zwischen den Darstellungsalternativen möglich ist. Nach einem Wechsel wird in Provop die jeweils neue Darstellung automatisch erzeugt. Dabei werden Usability-Aspekte berücksichtigt, wie etwa das Beibehalten der bisherigen Position der Elemente auf dem Bildschirm (vgl. „Mental Map“ [MELS95]).

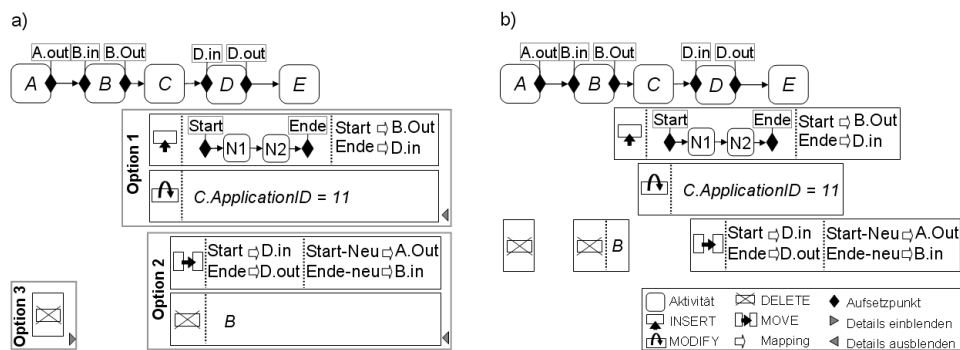


Abbildung 5: Positionierung von Optionen statisch (a) und verteilt (b)

4.2 Positionierung von Optionen in der Darstellung des Prozessmodells

Für Optionen muss angegeben werden, an welcher Position des Prozessmodells sie dargestellt werden sollen. Generell kann die Position einer Option abhängig von ihren Anpassungen gewählt werden; d.h. abhängig von den in den Anpassungen verwendeten Aufsetzpunkten bzw. IDs. Da in einer Option mehrere Anpassungen mit unterschiedlichen Aufsetzpunkten bzw. IDs definiert werden können, betrachten wir jeweils den „vordersten“ Start-Aufsetzpunkt aller Anpassungen einer Option bzw. den „ersten“ durch eine ID referenzierten Knoten im Basisprozess. Wir bezeichnen diese Position im Basisprozess als *Optionsstart*. Analog legt der „hinterste“ Ende-Aufsetzpunkt bzw. der „letzte“ durch eine ID referenzierte Knoten das *Optionsende* fest. Das Prozessfragment des Basisprozesses, das zwischen Optionsstart und Optionsende liegt, wird als *Optionsbereich* bezeichnet. Die folgenden Positionierungskonzepte für Optionen beziehen sich auf diese Definitionen:

- **Positionierung bzgl. Optionsstart und -ende:** Bei dieser Methode wird eine Option über die ganze Breite ihres Optionsbereichs, d.h. vom Optionsstart bis zum Optionsende, im Prozessmodell angezeigt. Dieser Ansatz stellt dem Modellierer den ganzen, von einer Option betroffenen Bereich im Basisprozess dar. Bei hoher Anzahl von Optionen mit großen Optionsbereichen kann dies dazu führen, dass das Prozessmodell unübersichtlich wird. Umgekehrt können Optionen bei sehr kleinen Optionsbereichen nur ausschnittsweise dargestellt werden, so dass z.B. mit Scrollbars gearbeitet werden muss.
- **Statische Positionierung bzgl. Optionsstart:** Bei der statischen Positionierung wird die Option am Optionsstart positioniert. Das Optionsende ist für die Positionierung nicht relevant. Abbildung 5a zeigt, dass Option 1 ihren Optionsstart am Ausgang von Knoten B des Basisprozesses hat. In diesem Ansatz sieht der Modellierer sehr gut, wann über die Verwendung einer Option entschieden wird. Jedoch kann, bei großer Verteilung der Anpassungen über den Basisprozess, die Übersichtlichkeit und Analysierbarkeit des Prozessmodells beeinträchtigt sein.

- **Dynamische Positionierung bzgl. Optionsstart:** Hier wird die Option solange am Optionsstart positioniert, wie dieser auf der Modellierungsoberfläche angezeigt wird. Scrollt der Modellierer durch das Prozessmodell und der Optionsstart ist nicht mehr im angezeigten Bereich sichtbar, bewegt sich die Option entsprechend mit und zwar solange bis der Optionsbereich nicht mehr im Bereich der Anzeige liegt. Gegebenenfalls kann sich die Option auch entsprechend ihrer Anpassungen bewegen, d.h. sie „springt“ beim Scrollen von Anpassung zu Anpassung. Die dynamische Positionierung der Optionen ermöglicht es dem Modellierer, alle Optionen des aktuell in der Modellierungsoberfläche betrachteten Bereichs des Basisprozesses zu überblicken. Die Anzeige kann aber „unruhig“ werden, wenn beim Scrollen ständig Optionen verschoben werden oder sogar „springen“.
- **Verteilte Positionierung:** Eine weitere Möglichkeit besteht darin, die verschiedenen Anpassungen einer Option verteilt im Prozessmodell anzuzeigen (vgl. Abbildung 5b). Der Modellierer sieht dann die jeweiligen Anpassungen dort, wo sie tatsächlich vorgenommen werden. Die Anzahl der dargestellten Objekte erhöht sich dadurch stark. Der Zusammenhang bzw. die Gruppierung der einzelnen Anpassungen zu Optionen geht in der Visualisierung verloren.

In Provop werden dem Modellierer alle vorgestellten Konzepte zur Positionierung von Optionen angeboten. Je nach persönlicher Präferenz und Anwendungsfall schaltet der Modellierer zwischen den Positionierungskonzepten um.

4.3 Darstellung des Ergebnismodells

Das Modellierungswerkzeug muss es ermöglichen, das durch Anwendung einer Menge von Optionen auf den Basisprozess entstehende Ergebnismodell anzuzeigen. Dazu muss das System dem Modellierer eine Möglichkeit zur Auswahl von Optionen bieten, die auf den Basisprozess angewendet werden sollen. Anschließend soll das Ergebnis der Anwendung berechnet und in einem Ergebnismodell dargestellt werden. Dabei müssen Konsistenzaspekte berücksichtigt werden, um sicherzustellen, dass die Optionen keine widersprüchlichen, redundanten oder nicht-kommutativen Anpassungen vornehmen. Die von Provop verfolgten Ansätze zur Darstellung eines Ergebnismodells werden exemplarisch anhand des in Abbildung 5 definierten Basisprozesses und seinen Optionen vorgestellt:

- **Normales Prozessmodell:** Das Ergebnismodell wird wie ein „normales“ Prozessmodell dargestellt. Das heißt, die durch die Anwendung von Optionen auf den Basisprozess manipulierten Prozesselemente werden nicht besonders gekennzeichnet.
- **Hervorheben der Optionen:** Das Darstellungsbeispiel aus Abbildung 6a zeigt das Ergebnismodell, das durch die gemeinsame Anwendung von Option 1 und Option 2 auf den Basisprozess entsteht. In dieser Darstellung kann der Modellierer erkennen, an welchen Stellen im Prozessmodell welche Änderungen durch welche Option vorgenommen worden sind. Dieser Ansatz ist aufgrund der wenigen zur Verfügung ste-

henden Hervorhebungsmerkmale (Rahmen, Farbe, Form, etc.) nur für eine begrenzte Zahl von Optionen geeignet.

- **Auswahl der hervorgehobenen Option:** In diesem Ansatz wird das Ergebnismodell als normales Prozessmodell angezeigt. Wird jedoch eine Option zur Anzeige ausgewählt, werden die von ihr geänderten Prozesselemente, soweit möglich, im Prozessmodell hervorgehoben. In Abbildung 6b ist Option 2 ausgewählt, daher wird die verschobene Aktivität *D* hervorgehoben. Durch diese Darstellung wird die Übersichtlichkeit des Prozessmodells erhöht, die modellierten Optionen sind aber schwerer vergleichbar.
- **Delta auf Basisprozess:** Bei den bisher diskutierten Formen der Darstellung von Ergebnismodellen werden nur die „sichtbaren“ Änderungen an Prozesselementen berücksichtigt. Das Entfernen von Prozesselementen hingegen ist im Ergebnismodell nicht mehr sichtbar. Um solche Veränderungen zu kennzeichnen, können entfernte Elemente in der Darstellung verbleiben und ausgegraut oder durchgestrichen dargestellt werden.
- **Markierung der Art der Änderung:** Merkmale wie Rahmen, Farbe, Form, etc. können dazu verwendet werden, um die durchgeführten Änderungsoperationstypen zu repräsentieren. Das Ergebnismodell aus Abbildung 6c zeigt z.B., dass von Aktivität *C* durch Änderung der Attribute eine neue Aktivität *C'* gebildet wurde. Parallel dazu wird das Prozessfragment mit den Aktivitäten *N1* und *N2* eingefügt. Aktivität *B* wurde durch Option 2 aus dem Basisprozess entfernt und erscheint im Ergebnismodell als durchgestrichener Knoten mit gestricheltem Rahmen. Als letzte Änderung wurde Aktivität *D* nach vorne verschoben. Dazu wurde die Symbolik für Entfernen und Einfügen wiederverwendet. Der Pfeil zwischen entfernter und eingefügter Aktivität zeigt, dass die Aktivität verschoben wurde. Die Aktivitäten *A* und *E* sind unverändert.

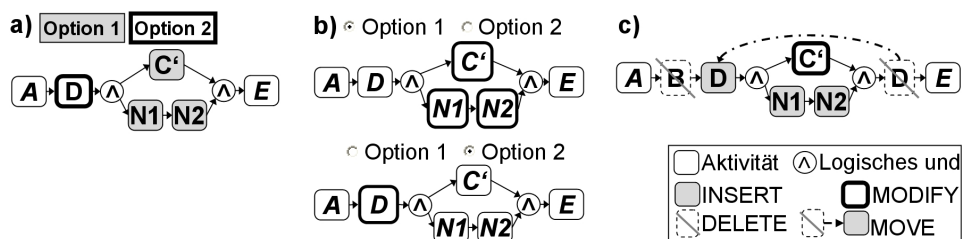


Abbildung 6: Unterscheidung der Optionen (alle (a) und ausgewählte (b)) bzw. der Änderungsoperationen (c)

Diese Darstellung erlaubt das Nachvollziehen der Änderungen, unterscheidet aber nicht mehr nach Optionen. Die Konzepte des Hervorhebens ausgewählter Optionen und die Unterscheidung der Änderungsoperationen sind jedoch kombinierbar.

5 Verwandte Arbeiten

Trotz hoher Praxisrelevanz werden Prozessvarianten in der Literatur bislang kaum betrachtet. Insbesondere wird die Modellierung mehrerer Prozessvarianten in einem Prozessmodell nicht ausreichend adressiert. Allerdings gibt es erste Ansätze, um den Umgang mit einer Vielzahl von Prozessmodellen handhabbar zu machen. So werden in [LS06] Prozessvarianten in einem Varianten-Repository verwaltet. Mit Hilfe von Such-Algorithmen können diejenigen Prozessvarianten gefunden werden, die der Struktur der Suchanfrage (in Form eines Prozessgraphen) ähneln. Im Gegensatz zu [LS06] werden in Provop die Prozessvarianten nicht separat modelliert oder getrennt vom Basisprozess verwaltet.

Ein Bereich, der sich wie Provop mit der Konstruktion von Modellen ausgehend von einem Prozessmodell beschäftigt, ist die Referenzmodellierung [Sch97, Bro03]. Referenzmodelle sind allgemein gültige Prozessmodelle, die für eine Klasse von Anwendungsfällen verwendet werden. Außerdem besitzen sie einen Empfehlungscharakter, d.h. „sie dienen als Ausgangslösung, aus der sich unternehmensspezifische Konkretisierungen ableiten lassen“ [Tho06]. Je nachdem wie der Modellierer den Basisprozess vorgibt, sind die Eigenschaften und Merkmale von Referenzmodellen auf unseren Ansatz übertragbar. So hat ein Standardprozess, der als Basisprozess dient, durchaus Empfehlungscharakter. Das Merkmal der Allgemeingültigkeit von Referenzprozessen kann in Provop für alle Prozessvarianten betrachtet werden, die in Summe eine Art „Klasse von Anwendungsfällen“ repräsentieren. In manchen Veröffentlichungen zur Referenzmodellierung wird gefordert, dass ein Basisprozess für mindestens einen Anwendungsfall unverändert eingesetzt werden kann. Durch die verschiedenen Ansätze zur Festlegung des Basisprozesses in Provop ist dies nicht immer gegeben. So sind Basisprozesse, die als Schnittmenge aller Prozessvarianten definiert wurden oder für einen „minimalen Modellierungsaufwand“ optimiert sind (vgl. Abschnitt 2.1), konstruierte Prozesse ohne spezifischen Anwendungsfall.

Ein konkreter Ansatz zur Referenzmodellierung bietet das Konzept der konfigurierbaren Ereignis-Prozess-Ketten (C-EPCs) [RvdA07]. C-EPCs ermöglichen das Erstellen von Varianten auf Grundlage eines konfigurierbaren Referenzprozesses. Hier können Funktionen bei der Variantenbildung so konfiguriert werden, dass sie entweder fester Bestandteil der Prozessvariante sind, ausgelassen werden müssen oder optional sind. Dazu werden die Funktionen mit Informationen hinterlegt, die dem Modellierer bei der Variantenbildung unterstützen. Der Ansatz beschränkt sich zur Zeit noch auf den Kontrollfluss, daher wird die Anpassung von Attributen, wie in Provop, nicht berücksichtigt. Auch das Hinzufügen und Verschieben von Funktionen ist bei C-EPCs nicht möglich.

In der Softwareentwicklung nehmen Varianten traditionell eine wichtige Rolle ein. Neben grundlegenden Charakterisierungen von Variabilität [BB01] werden im Kontext von Software-Architekturen und Software-Produktfamilien auch Prozessvarianten behandelt [HP03, BGGB01]. So finden sich z.B. bei PESOA [BBG⁺05, PSWW05] Konzepte zur Modellierung von Prozessvarianten in einem UML-Modell. Es werden verschiedene Variabilitätsmechanismen (z.B. Vererbung, Parametrisierung, Erweiterungspunkte, etc.) vorgestellt und beispielhaft in verschiedenen Modelltypen angewendet. Die Variabilitätsmechanismen werden durch entsprechende UML-Konstrukte direkt in einem Prozess modelliert. Im Gegensatz zu Provop findet keine logische oder modellierungstechnische Tren-

nung von Basisprozess und Prozessvarianten statt. Auch beschränken sich in Provop die zusätzlich notwendigen Konstrukte auf Optionen. Die unterschiedlichen Variabilitätsmechanismen werden durch intuitive Änderungsoperationen beschrieben. Dadurch wird die Darstellung von Basisprozess und Optionen insgesamt übersichtlicher und verständlicher, und die Komplexität für den Modellierer reduziert.

6 Zusammenfassung und Ausblick

Die Modellierung von Prozessen erfordert in der Praxis häufig die Abbildung von Varianten. Heutige Modellierungswerkzeuge erlauben jedoch nur das Ausmodellieren dieser Varianten in separaten Prozessmodellen oder die Modellierung mittels bedingter Verzweigungen in einem einzigen Prozessmodell. Die Nachteile dieser Ansätze sind hohe Modellierungs- und Wartungsaufwände, Fehleranfälligkeit sowie implizite Variantenbeschreibungen.

Der im vorliegenden Beitrag vorgestellte Provop-Ansatz ermöglicht die explizite und zusammenhängende Modellierung der Varianten eines Prozessmodells, ohne den Nachteil, dass die Information zu den Varianten in der Ablauflogik versteckt sind. Dazu verwenden wir Optionen in denen jeweils eine Menge von Anpassungen des Basisprozesses beschrieben ist. Diese Anpassungen wirken auf den Basisprozess durch die Änderungsoperationen INSERT, DELETE, MOVE und MODIFY. Mit Hilfe von Aufsetzpunkten bzw. Prozesselement-IDs können die von einer Änderung betroffenen Elemente des Basisprozesses referenziert werden. Schließlich erlaubt die Abbildung logischer Aufsetzpunkte der Anpassungen auf die konkreten Aufsetzpunkte im Basisprozess die generische Beschreibung und Wiederverwendung von Optionen zur Erstellung einer Prozessvariante in anderen Basisprozessen.

Neben Grundlagen unseres Modellierungsansatzes für Prozessvarianten haben wir diskutiert, wie Optionen in einem Prozessmodell für Benutzer dargestellt werden können. Außerdem haben wir die Möglichkeiten zur Positionierung von Optionen in der Modellierungsoberfläche analysiert, um so die Vergleichbarkeit und Übersichtlichkeit der Optionen zu verbessern. Um dem Benutzer maximale Flexibilität einzuräumen, müssen alle vorgestellten Konzepte individuell ausgewählt und kombiniert werden können.

Zusätzlich zu den vorgestellten Modellierungsthemen werden in Provop weitere Phasen des Prozesslebenszyklus untersucht [HBR07]. Zukünftig Themen sind z.B. die Überprüfung der Konsistenz der modellierten und zu kombinierenden Optionen, sowie die Erweiterung des Ansatzes um den Aspekt der Kontextsensitivität. Ziel ist ferner, die in Provop modellierten Prozessvarianten mit Hilfe von Prozess-Management-Technologie aktiv steuern zu können. Eine umfassende Validation unseres Ansatzes mit Verfügbarkeit entsprechender Modellierungswerkzeuge steht ebenfalls noch aus.

Literatur

- [BB01] F. Bachmann und L. Bass. Managing variability in software architectures. In *Proc. of the 2001 Symp. on Software Reusability*, Seiten 126–132, New York, 2001. ACM Press.
- [BBG⁺05] J. Bayer, W. Buhl, C. Giese, T. Lehner, A. Ocampo, F. Puhlmann, E. Richter, A. Schnieders, J. Weiland und M. Weske. PESOA - Process Family Engineering - Modeling variant-rich processes. Bericht 18/2005, Hasso-Plattner-Institut, Potsdam, 2005.
- [BGGB01] M. Becker, L. Geyer, A. Gilbert und K. Becker. Comprehensive Variability Modeling to Facilitate Efficient Variability Treatment. In *Proc. of the 4th Int. Workshop od Product Family Engineering*, 2001.
- [Bro03] J. Brocke. *Referenzmodellierung - Gestaltung und Verteilung von Konstruktionsprozessen*. Dissertation, Westfälische Wilhelms-Uni Münster, 2003.
- [Gad05] A. Gadatsch. *Grundkurs Geschäftsprozess-Management*. Vieweg Verlag, 2005.
- [HBR07] A. Hallerbach, T. Bauer und M. Reichert. Managing Process Variants in the Process Life Cycle. Bericht TR-CTIT-07-87, Uni Twente, NL, 2007. ISSN 1381-3625.
- [HBR08] A. Hallerbach, T. Bauer und M. Reichert. Anforderungen an die Modellierung und Darstellung von Prozessvarianten. In *Datenbank Spektrum*. Datenbank Spektrum, 2008.
- [HP03] G. Halmans und K. Pohl. Communicating the Variability of a Software-Product Family to Customers. *Software and System Modeling*, 2(1):15–36, 2003.
- [IBM07] IBM Corporation. *IBM WebSphere Business Modeller, Version 6.0.2*, 2007.
- [IDS06] IDS Scheer AG. *ARIS Platform Method 7.0*, Oktober 2006.
- [LS06] R. Lu und S. Sadiq. On Managing Process Variants as an Information Resource. Bericht No. 464, School of Information Technology & Electrical Engineering and University of Queensland, Brisbane, Juni 2006.
- [MELS95] K. Misue, P. Eadest, W. Lai und K. Sugiyama. Layout Adjustment and the Mental Map. *Visual Languages and Computing*, 6:183–210, 1995.
- [Mey96] J. Meyer. Anforderungen an zukünftige Workflow-Management-Systeme: Flexibilisierung, Ausnahmebehandlung und Dynamisierung - Erörterung am Beispiel medizinisch-organisatorischer Abläufe. Diplomarbeit, Uni Ulm, 1996.
- [PSWW05] F. Puhlmann, A. Schnieders, J. Weiland und M. Weske. PESOA - Variability Mechanisms for Process Models. Bericht 17/2005, Hasso-Plattner-Institut, Potsdam, 2005.
- [RvdA07] M. Rosemann und W.M.P. van der Aalst. A configurable reference modelling lanugage. *Information Systems*, 32:1–23, 2007.
- [Sch97] R. Schütte. *Grundsätze ordnungsgemäßer Referenzmodellierung: Konstruktion konfigurations- und anpassungsorientierter Modelle*. Dissertation, Uni Münster, 1997.
- [Sch98] A.-W. Scheer. *ARIS- Modellierungsmethoden, Metamodelle, Anwendungen*. Springer-Verlag, 1998. Vierte Auflage.
- [Tho06] O. Thomas. Das Referenzmodellverständnis in der Wirtschaftsinformatik: Historie, Literaturanalyse und Begriffsexplikation. Bericht 187, Institut für Wirtschaftsinformatik im Deutschen Forschungszentrum für Künstliche Intelligenz, Januar 2006.
- [VDA05] VDA Recommendation 4965 T1. Engineering Change Management (ECM) - Part 1: Engineering Change Request (ECR) Version 1.1, Dezember 2005.